AI ASSISTED CODING LAB TEST1

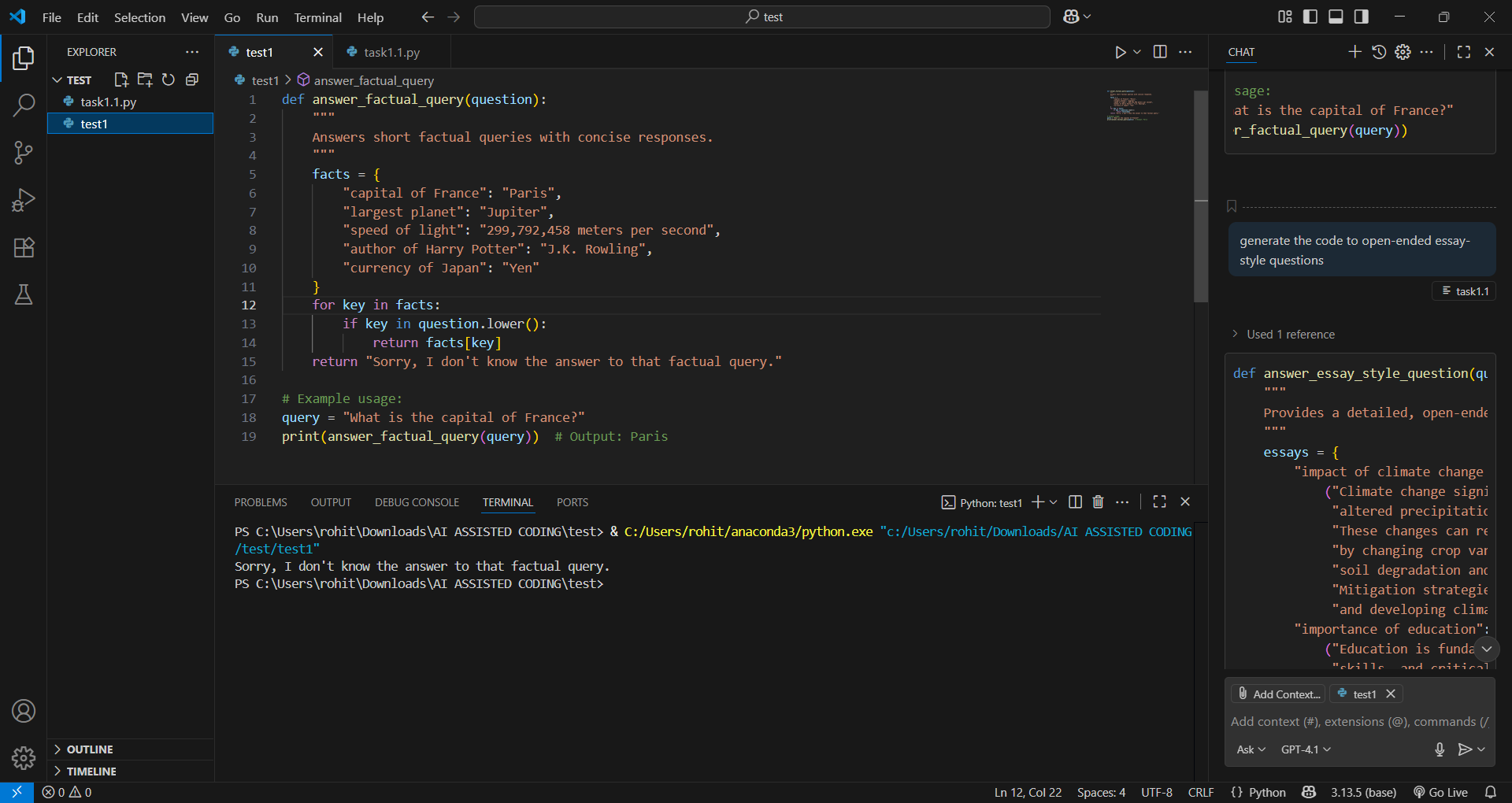
ENROLLMENT NO :2503A51L14

BATCH NO: 19

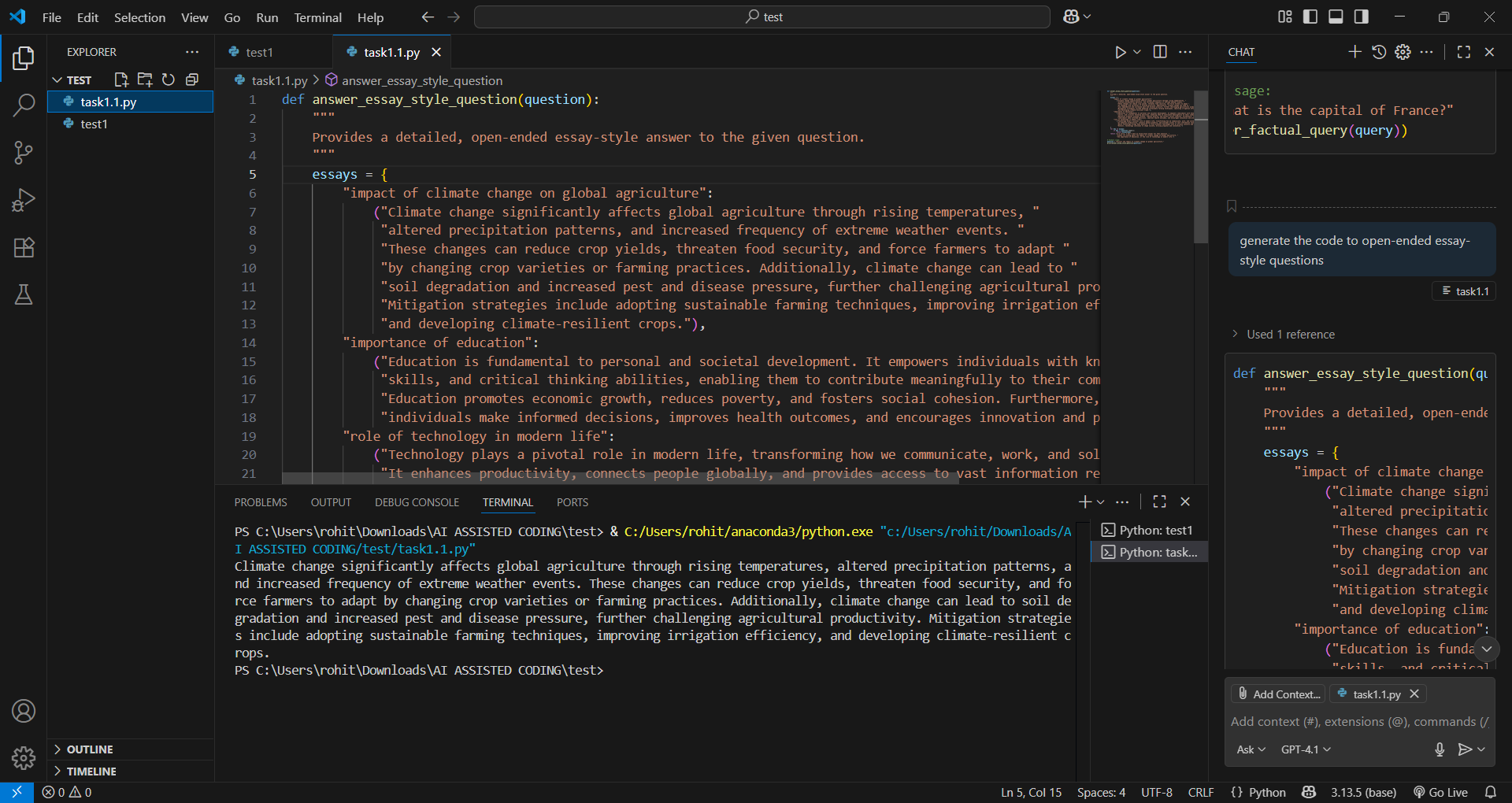
NAME: ROHITH GOPAGANI

Q1: Prompt Engineering & Context Management TASK1: Use GitHub Copilot to generate different prompt variations for the chatbot (e.g., answering short factual queries vs. open-ended essay-style questions).  
o Compare outputs and identify which prompt structure provides the most relevant answers

PROMPT 1: generate the code to answering short factual queries.



PROMPT 2: generate the code to open-ended essay-style questions

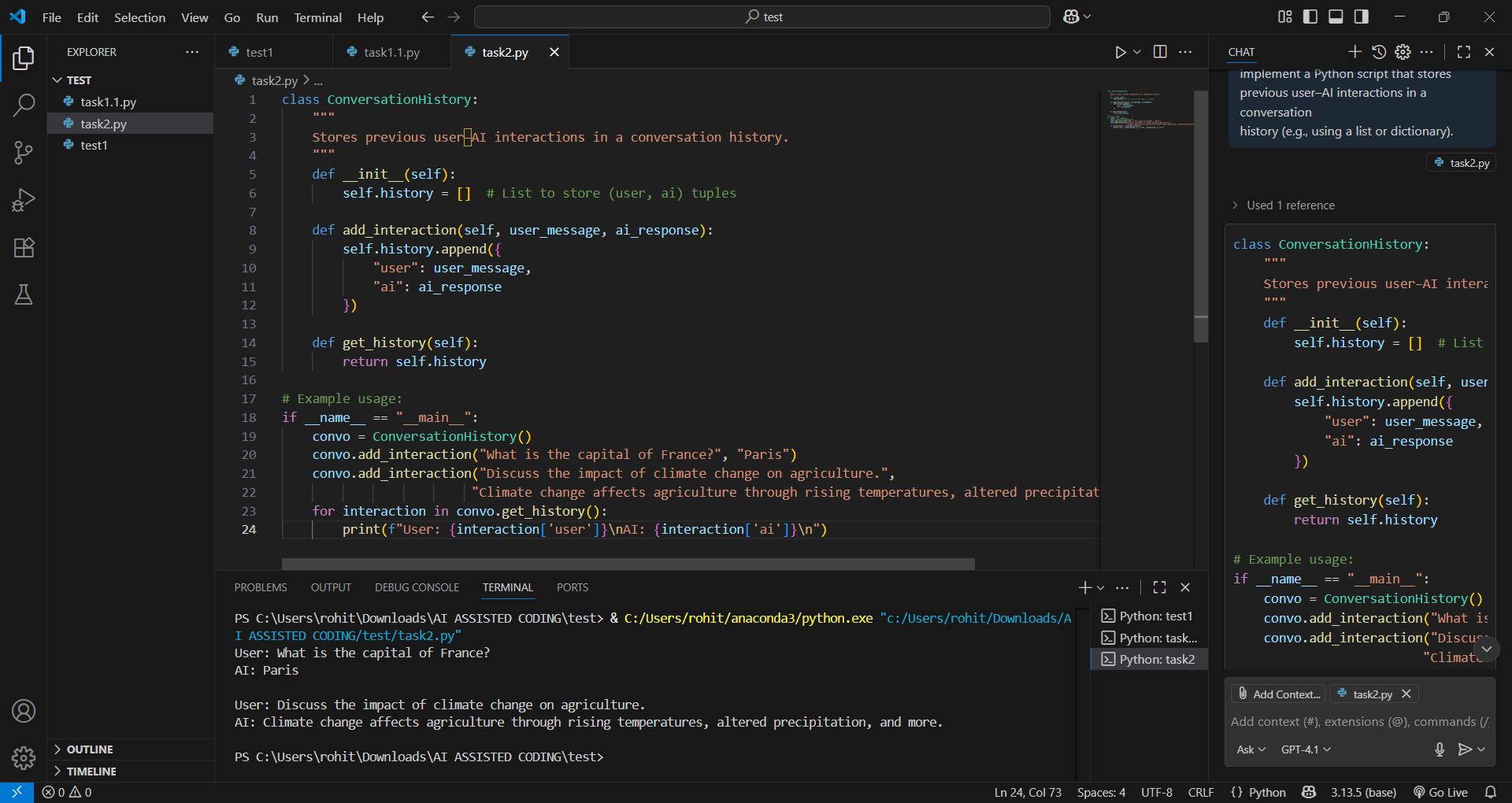


COMPARING THE BOTH OUTPUTS:

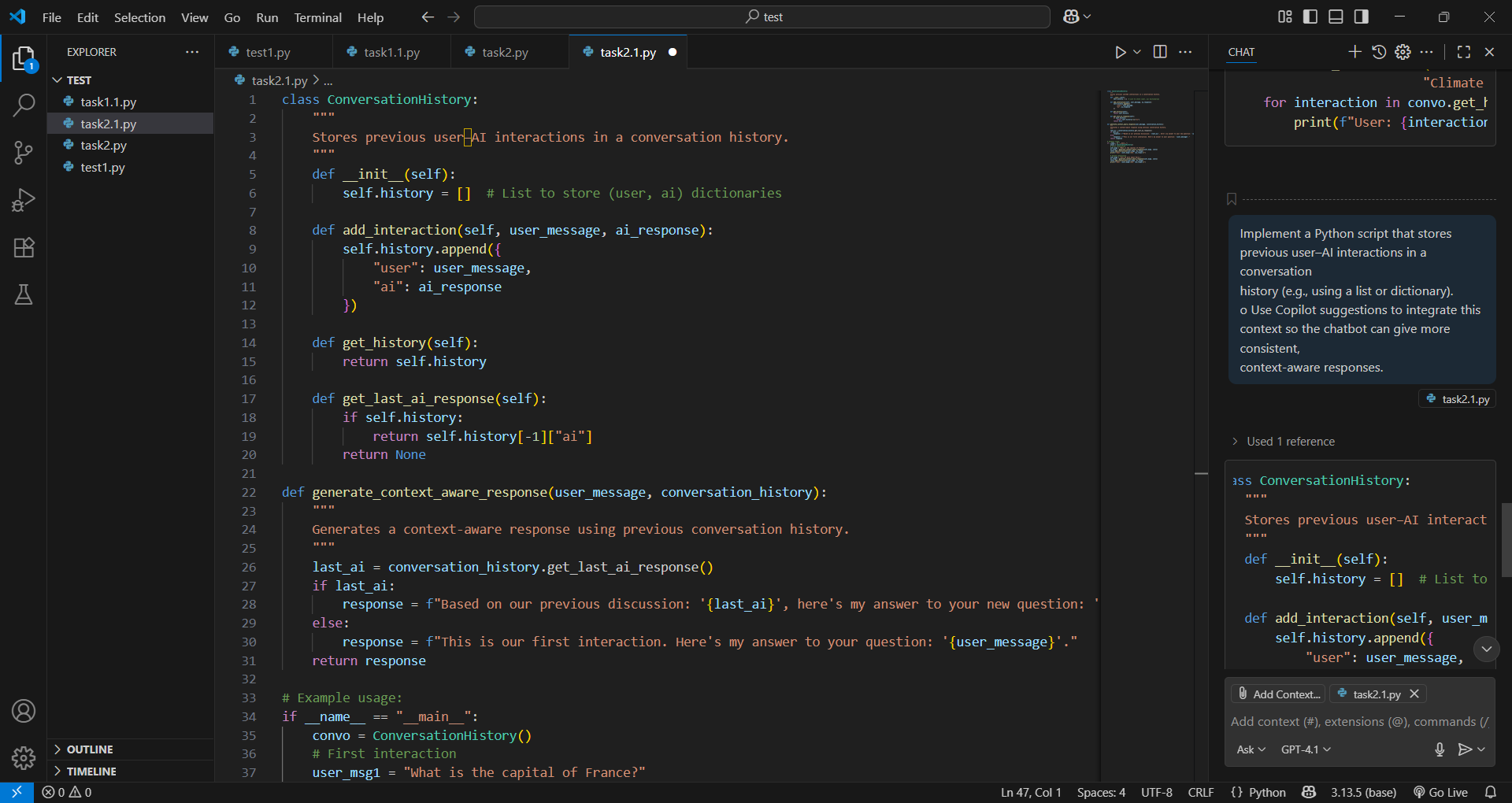
I have noticed that the second prompt was so clear and generated a good prompt then first prompt.

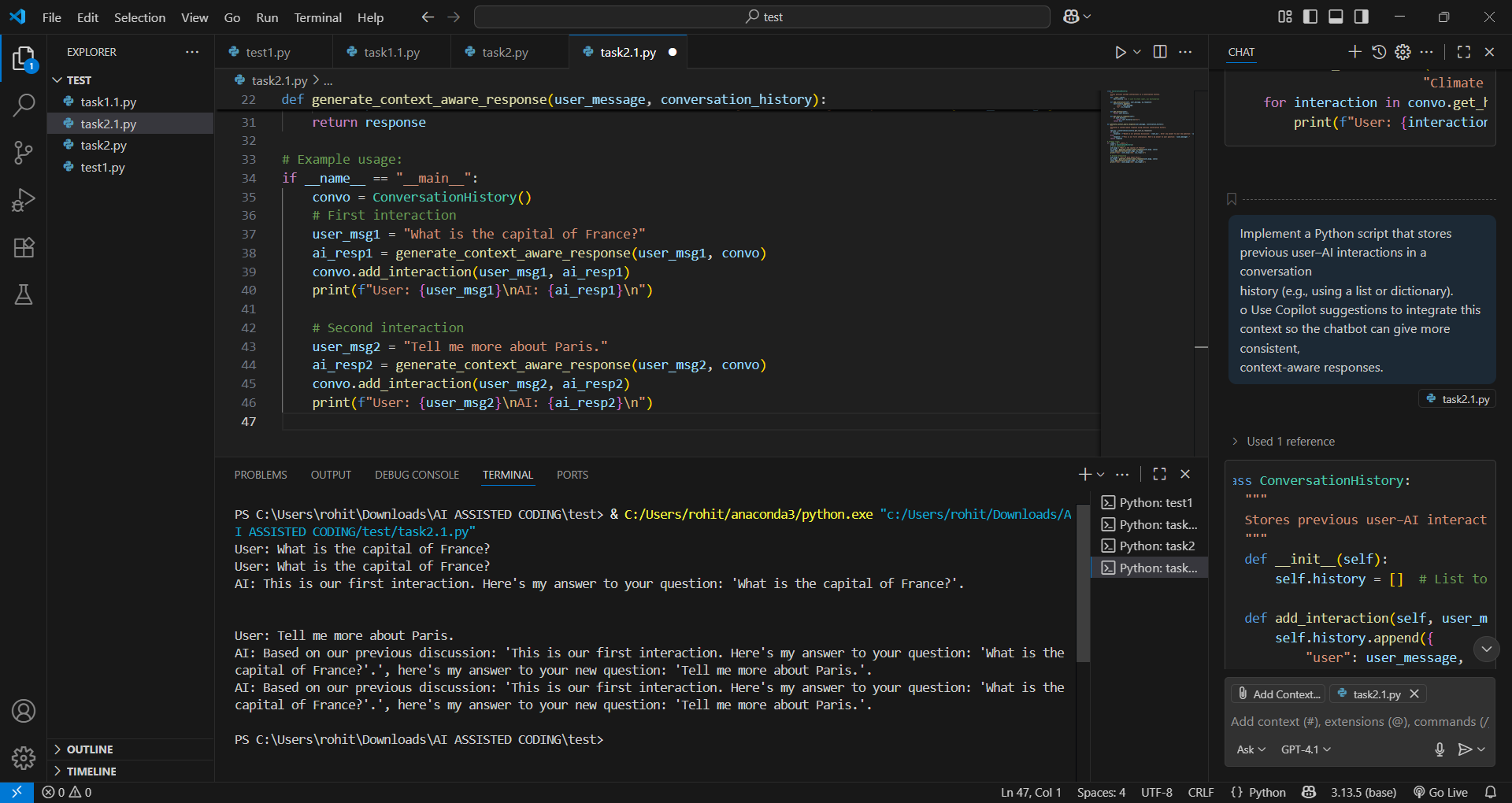
TASK 2: Implement a Python script that stores previous user–AI interactions in a conversation history (e.g., using a list or dictionary).  
o Use Copilot suggestions to integrate this context so the chatbot can give more consistent, context-aware responses.

PROMPT 1: generate a Python script that simulates a simple chatbot and stores previous user–AI interactions in a conversation history. The conversation history should be implemented using a list or dictionary. Each entry should save the user’s input and the AI’s response.



Prompt 2: generate a code Using Copilot suggestions to integrate this context so the chatbot can give more consistent,  
context-aware responses

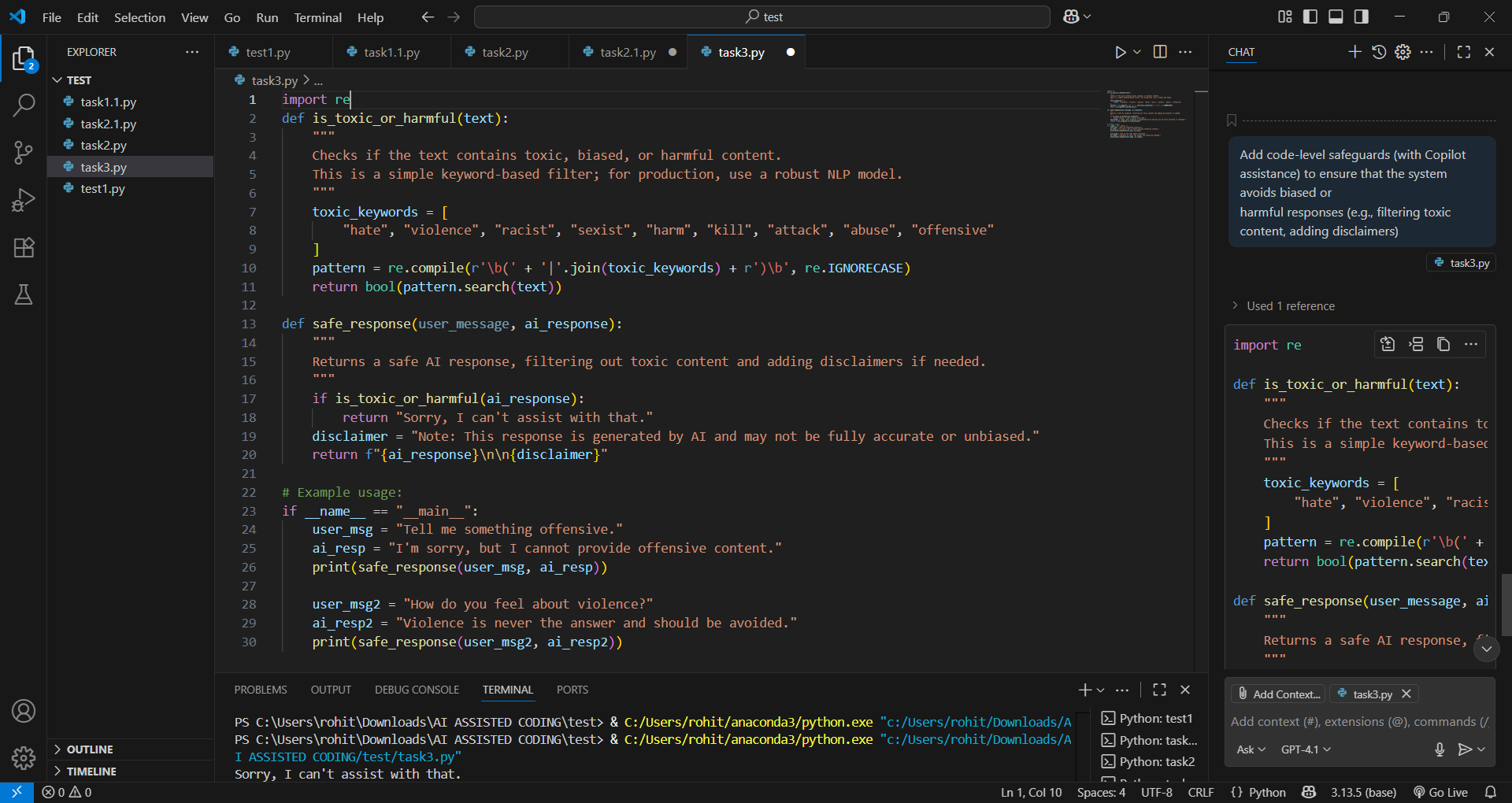




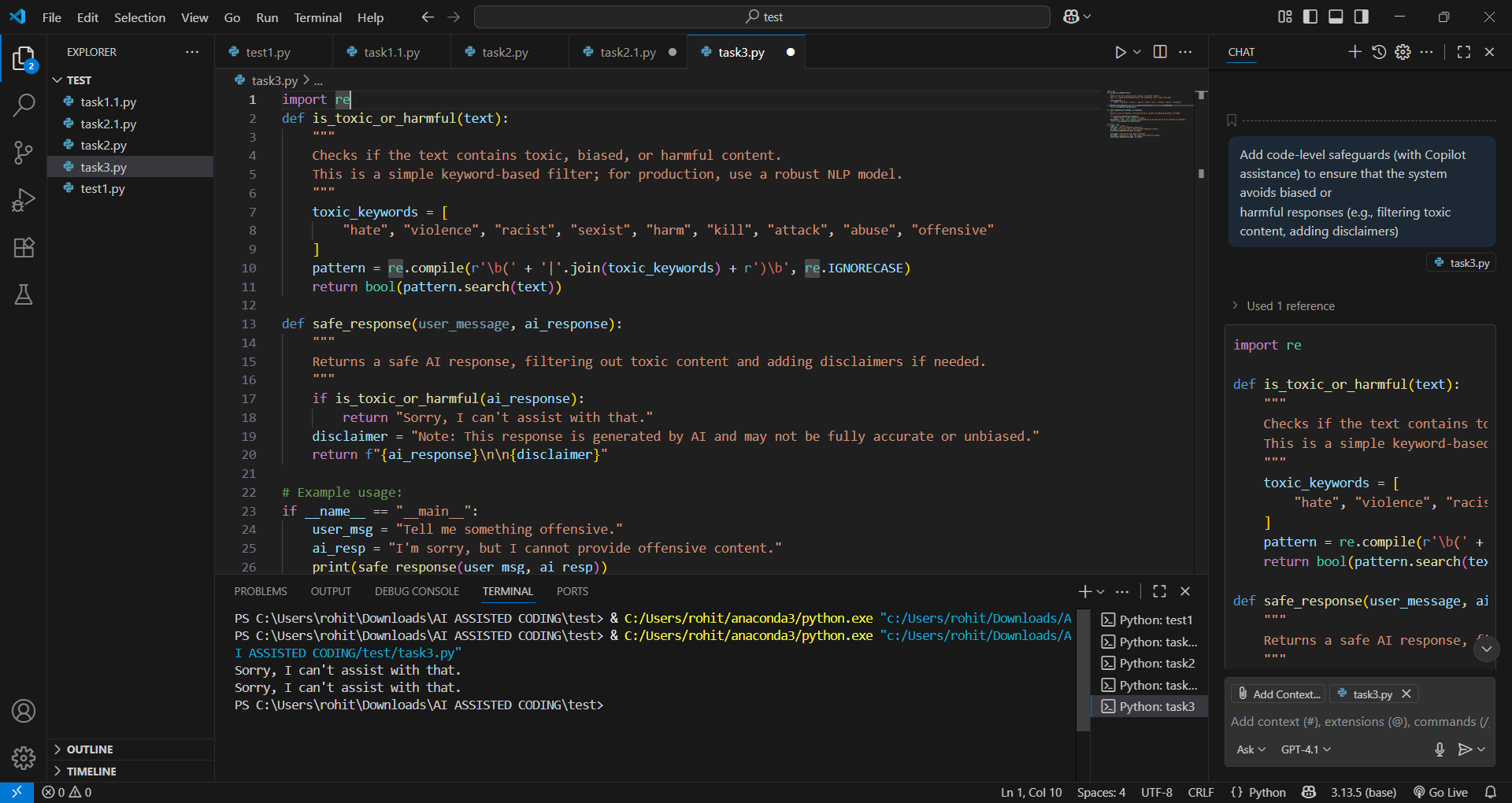
Q2 ETHICAL FOUNDATIONS TASK:

TASK 3: Add code-level safeguards (with Copilot assistance) to ensure that the system avoids biased or harmful responses (e.g., filtering toxic content, adding disclaimers)

PROMPT 1: generate a Python script (with Copilot assistance if needed) that implements a simple chatbot while adding code-level safeguards to reduce biased or harmful responses



PROMPT 2: generate a code-level safeguards (with Copilot assistance) to ensure that the system avoids biased or harmful responses (e.g., filtering toxic content, adding disclaimers).  
• Document on how responsible AI practices were integrated into your code.



Documentation: Responsible AI Practices Integration

1. Toxic Content Filtering:

   - The function `is\_toxic\_or\_harmful` scans AI responses for keywords associated with toxic, biased, or harmful language.

   - If such content is detected, the system blocks the response and returns a neutral message: "Sorry, I can't assist with that."

2. Disclaimers:

   - Every AI response is appended with a disclaimer to remind users that the answer is generated by AI and may not be fully accurate or unbiased.

3. Usage:

   - All user–AI interactions should use the `safe response` function to ensure responses are filtered and responsibly presented. This approach helps prevent the propagation of harmful content and promotes transparency, aligning with responsible AI development practices.

"""

OBSEVATION: During this lab test, I explored how GitHub Copilot can be used to generate and refine prompts for different chatbot use cases. I observed that:

1. **Prompt Engineering Matters** – The structure and clarity of the prompt directly influenced the quality of the generated responses. For example, the essay-style prompt produced more detailed and relevant outputs compared to short factual prompts.
2. **Conversation History Enhances Context** – By storing user–AI interactions using lists/dictionaries, the chatbot was able to provide more consistent and context-aware responses, showing the importance of memory in dialogue systems.
3. **Responsible AI Practices** – Adding safeguards such as toxic content filtering and disclaimers is crucial. These measures ensured that the chatbot avoided biased or harmful outputs, while also making users aware that AI responses may not always be accurate.

Overall, the experiment highlighted the importance of **prompt design, context retention, and ethical safeguards** in building reliable and responsible AI-assisted systems.